CLAIMS

What is claimed is:

1. A method of processing messages comprising:

generating source code in a program language;

compiling the source code; and

using the compiled source code to transform a payload message between the program language and an online transaction processing (OLTP) language, the source code being generated automatically.

2. The method of claim 1, further including:

receiving a markup language document, the markup language document being compliant with message rules associated with the OLTP language; and

compiling the markup language document into the source code, the source code enabling transformation of the message between the program language and the OLTP language.

3. The method of claim 2, wherein the message rules define markup tags for describing components of the message.

4. The method of claim 3, wherein the message is a request message from an application to an OLTP host.

5. The method of claim 3, wherein the message is a response message from an OLTP host to an application.

6. The method of claim 3, wherein the markup language is an extensible markup language.

7. The method of claim 2, further including compiling the markup language document based on stylesheet data, the stylesheet data representing a program template for the message in the program language.

8. The method of claim 7, further including using a stylesheet parser to compile the markup language document.

9. The method of claim 2, further including receiving and compiling a plurality of markup language documents for a plurality of message types.

10. The method of claim 1, wherein the program language is an object-oriented program language.

11. The method of claim 10, further including:

receiving one or more name-value pairs for the message according to the object-oriented program language;

generating a byte array according to the OLTP language; and

transmitting the byte array to an application.

12. The method of claim 10, further including:

receiving a byte array for the message according to the OLTP language;

generating one or more name-value pairs according to the object-oriented program language; and

transmitting the name-value pairs to an application.

13. The method of claim 10, wherein the object-oriented program language is a Java language.

14. The method of claim 1, wherein the OLTP language is a gaming system language.

15. A method of generating source code comprising:

receiving a markup language document, the markup language document being compliant with message rules associated with an online transaction processing (OLTP) language; and compiling the markup language document into source code, the source code enabling transformation of a payload message between a program language and the OLTP language.

16. The method of claim 15, wherein the message rules define markup tags for describing components of the message.

17. The method of claim 16, wherein the message is a request message from an application to an OLTP host.

18. The method of claim 16, wherein the message is a response message from an OLTP host to an application.

19. The method of claim 16, wherein the markup language is an extensible markup language.

20. The method of claim 15, further including compiling the markup language document based on stylesheet data, the stylesheet data representing a program template for the message in the program language.

21. The method of claim 20, further including using a stylesheet parser to compile the markup language document.

22. The method of claim 15, further including receiving and compiling a plurality of markup language documents for a plurality of message types.

23. A method of processing messages comprising:

receiving one or more extensible markup language (XML) documents, the XML documents being compliant with message rules associated with an online transaction processing (OLTP) language, the OLTP language being a gaming system language and the message rules defining markup tags for describing components of the message;

compiling the XML documents into source code in an object-oriented program language based on stylesheet data, the stylesheet data representing a program template for a payload message in the program language, the source code enabling transformation of the message between the program language and the OLTP language;

compiling the source code;

using the compiled source code to transform a message between the program language and the OLTP language, the source code being generated automatically; and

repeating the receiving and compiling of XML documents and the compiling of source code for a plurality of message types.

24. The method of claim 23, further including using a stylesheet parser to compile the XML documents.

25. The method of claim 23, further including:

receiving one or more name-value pairs for the message according to the program language;

generating a byte array according to the OLTP language; and

transmitting the byte array to an application.

26. The method of claim 23, further including:

receiving a byte array for the message according to the OLTP language;

generating one or more name-value pairs according to the program language; and

transmitting the name-value pairs to an application.

27. The method of claim 23, wherein the object-oriented program language is a Java language.

28. A message processing architecture comprising:

a development module, the development module to generate source code in a program language;

a program compiler, the program compiler to compile the source code; and a message parser, the message parser to transform a payload message between the program language and an online transaction processing (OLTP) language, the development module to generate the source code automatically.

29. The system of claim 28, wherein the development module is to receive a markup language document and compile the markup document into the source code, the markup language document to be compliant with message rules associated with the OLTP language and the source code to enable transformation of the message between the program language and the OLTP language.

30. The system of claim 29, wherein the message rules define markup tags for describing components of the message.

31. The system of claim 29, wherein the development module further includes a stylesheet parser, the development module to use the stylesheet parser to compile the markup language document based on stylesheet data.

32. A machine readable medium comprising a set of stored instructions capable of being executed by a processor to:

receive a markup language document, the markup language document to be compliant with message rules associated with an online transaction processing (OLTP) language; and

compile the markup language document into source code, the source code to enable transformation of a message between a program language and the OLTP language.

33. The medium of claim 32, wherein the message rules define markup tags for describing components of the message.

34. The medium of claim 32, wherein the instructions are further capable of being executed to compile the markup language document based on stylesheet data, the stylesheet data to represent a program template for the message in the program language.

35. The medium of claim 32, wherein the instructions are further capable of being executed to receive and compile a plurality of markup language documents for a plurality of message types.